

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 07-006110

(43)Date of publication of application : 10.01.1995

(51)Int.Cl.

G06F 13/00

G06F 15/16

(21)Application number : 05-143184

(71)Applicant : HITACHI LTD

(22)Date of filing : 15.06.1993

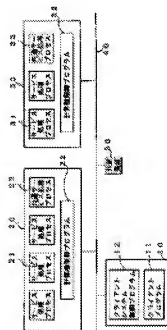
(72)Inventor : FUJIWARA HIROKAZU  
IWASAKI MOTOAKI  
IGARASHI AKIO  
HASEGAWA HIROYUKI  
HASHIMOTO TADASHI

## (54) METHOD FOR REDUCING COMMUNICATION OVERHEAD IN DISTRIBUTED PROCESSING SYSTEM

### (57)Abstract:

**PURPOSE:** To prevent the generation of communication overhead following service request transfer between plural computers and the reduction of processing performance of the whole system by transmitting the same request from the same client to a computer having practically processed the request based upon identification information.

**CONSTITUTION:** At the time of knowing the return of a result from a computer 30 different from a computer 20 having sent a service request, a client system control program 12 updates service corresponding to the result and an item in a correspondence table stored in the computer 30 and returns the updated results to a client program 11. At the time of receiving the same service request again from the program 11, the control program 12 transmits the service request of the program 11 to a computer control program 32 for the computer 30 based upon the stored service and correspondence table of the computer 30 without transmitting the service request of the program 11 to the same computer 20 as that of the preceding service request.



(51) Int. Cl. <sup>6</sup>	識別記号	序内整理番号	F I	技術表示箇所
G 0 6 F 13/00	3 5 7 Z	7368-5B		
15/16	3 8 0 Z	7429-5L		

審査請求 未請求 請求項の数 2 O L (全 13 頁)

(21) 出願番号 特願平5-143184

(22) 出願日 平成5年(1993)6月15日

(71) 出願人 000005108  
株式会社日立製作所  
東京都千代田区神田駿河台四丁目6番地

(72) 発明者 藤原 弘和  
神奈川県横浜市戸塚区戸塚町5030番地 株式会社日立製作所ソフトウェア開発本部内

(72) 発明者 岩崎 元昭  
神奈川県横浜市戸塚区戸塚町5030番地 株式会社日立製作所ソフトウェア開発本部内

(72) 発明者 五十嵐 明夫  
神奈川県横浜市戸塚区戸塚町5030番地 株式会社日立製作所ソフトウェア開発本部内

(74) 代理人 弁理士 秋田 収喜

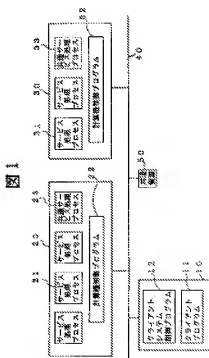
最終頁に続く

## (54) 【発明の名称】 分散処理システムの通信オーバーヘッド低減方法

## (57) 【要約】

【目的】 動的負荷分散環境下のクライアント・サーバ型分散処理システムにおいて、重負荷計算機へのクライアントの要求を避け、計算機間の要求転送の発生に伴う通信オーバーヘッドを低減し、システムの性能向上を図る。

【構成】 動的負荷分散環境下のクライアント・サーバ型分散処理システムにおいて、クライアントが、ある計算機へサービス要求を出した時に、実際に要求を処理した計算機を識別するための識別情報を含んだ結果を、クライアントに返送することにより、クライアントが実際に要求を処理した計算機を認識できるようにし、以後の同一の要求は実際に要求を処理した計算機へ送る。



【特許請求の範囲】

【請求項1】 複数の計算機をネットワークで接続し、各計算機にかかる負荷を監視し、負荷のバランスを保つ動的負荷分散環境下で、各計算機を呼び出して1つの処理を実行していくクライアント・サーバ型分散処理システムにおいて、クライアントが任意の計算機へサービス要求を出した時に、実際にそのサービス要求を処理した計算機を識別する識別情報を含んだ処理結果をクライアントに返送しておき、前記識別情報に基づき、以後の同一クライアントの同一の要求は実際に要求を処理した計算機へ送信することを特徴とする分散処理システムの通信オーバーヘッド低減方法。

【請求項2】 複数の計算機をネットワークで接続し、各計算機にかかる負荷を監視し、負荷のバランスを保つ動的負荷分散環境下で、各計算機を呼び出して1つの処理を実行していくクライアント・サーバ型分散処理システムにおいて、共通サービス処理プロセスが要求を受け取った際に、共通サービス処理プロセス自身が各計算機からの共通サービスへの要求を出す頻度を求め、前記頻度に基づき、他の計算機へ共通サービスプロセスを再配置すべきか否かを判断し、その判断結果に基づき、他の計算機へ共通サービスプロセスを再配置することを特徴とする分散処理システムの通信オーバーヘッド低減方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、複数の計算機をネットワークで接続し、各計算機にかかる負荷を監視し、負荷のバランスを保つ動的負荷分散環境下で、各計算機を呼び出して1つの処理を実行していくクライアント・サーバ型分散処理システムにおける、負荷分散時あるいは共通サービスへの要求時の通信オーバーヘッドの低減に関するものである。

【0002】

【従来の技術】例えば、特開平2-280266号に記載されているように、各計算機の負荷が均等になるように短時間で各計算機における負荷の割当を行うようにして、処理能力を向上させたクライアント・サーバ型分散処理システムが知られている。

【0003】従来のクライアント・サーバ型分散処理システムにおいて、動的負荷分散が行っている場合に、クライアントからの要求を受け付けた計算機が他の計算機よりも重い負荷がかかっていたとすると、クライアントからの要求を受け付けた計算機は、他の軽負荷の計算機へのクライアントからの要求を転送し、転送を受けた他の軽負荷の計算機がクライアントからの要求を処理することになる。

【0004】しかし、クライアントでは、どの計算機から結果が返ってきたのか不明であり、クライアントが再び同一の要求を出す時は、前回の要求先と同じ計算機へ

2

その要求を送ることになる。そして、その要求を受け取った計算機が、再び他の計算機よりも重い負荷がかかっていたとすると、再び負荷分散が必要になり、クライアントからの要求を受け付けた計算機は、他の軽負荷の計算機へのクライアントからの要求を転送するということが繰り返し発生することになる。

【0005】また、従来のクライアント・サーバ型分散処理システムにおいて、共通サービスに関しては、動的負荷分散の対象となっておらず、共通サービスを起動させた計算機へ要求が集まり、他の計算機からの要求が多くなると、そのための通信オーバーヘッドが増加することになる。

【0006】

【発明が解決しようとする課題】本発明者は、複数の計算機をネットワークで接続し、各計算機にかかる負荷を監視し、負荷のバランスを保つ動的負荷分散環境下で、各計算機を呼び出して1つの処理を実行していくクライアント・サーバ型分散処理システムにおいて、従来例について検討した結果、以下の問題点を見出した。

【0007】(a) 前記の従来技術では、重負荷な計算機へ要求が繰り返されることに伴う要求受付というさらなる負荷増と、要求を他の計算機へ転送するための通信オーバーヘッドの発生について考慮されておらず、それによりシステムの処理性能が低下するという問題があった。

【0008】(b) 前記の従来技術では、共通サービスへ要求する際の通信オーバーヘッドの発生について配慮されておらず、それによりシステムの処理性能が低下するという問題があった。

【0009】本発明は、前記問題点を解決するためになされたものであって、本発明の目的は、複数の計算機をネットワークで接続し、各計算機にかかる負荷を監視し、負荷のバランスを保つ動的負荷分散環境下で、各計算機を呼び出して1つの処理を実行していくクライアント・サーバ型分散処理システムにおいて、クライアントの要求先計算機についても動的に変更することにより重負荷計算機へのクライアントの要求を避け、計算機間の要求転送の発生に伴う通信オーバーヘッドを低減し、システムの性能向上を図ることにある。

【0010】本発明の他の目的は、複数の計算機をネットワークで接続し、各計算機にかかる負荷を監視し、負荷のバランスを保つ動的負荷分散環境下で、各計算機を呼び出して1つの処理を実行していくクライアント・サーバ型分散処理システムにおいて、共通サービス処理プロセス自身に効率の良い計算機を判断させ、共通サービス処理プロセスを再配置することにより、計算機間の共通サービスへの要求に伴う通信オーバーヘッドを低減し、システムの性能向上を図ることにある。

【0011】本発明の前記並びにその他の目的及び新規な特徴は、本明細書の記載及び添付図面によって明らか

3

になるであろう。

【0012】

【課題を解決するための手段】前記目的を達成するために、本発明は、複数の計算機をネットワークで接続し、各計算機にかかる負荷を監視し、負荷のバランスを保つ動的負荷分散環境下で、各計算機を呼び出して1つの処理を実行していくクライアント・サーバ型分散処理システムにおいて、クライアントが任意の計算機へサービス要求を出した時に、実際にそのサービス要求を処理した計算機を識別する識別情報を含んだ処理結果をクライアントに返送しておき、前記識別情報に基づき、以後の同一クライアントの同一の要求は実際に要求を処理した計算機へ送信することにより、負荷分散時の計算機間の通信オーバーヘッドを低減させることを特徴とする。

【0013】また、他の目的を達成するために、共通サービス処理プロセスが要求を受け取った際に、共通サービス処理プロセス自身が各計算機からの共通サービスへの要求を出す頻度を求め、前記頻度に基づき、他の計算機へ共通サービスプロセスを再配置すべき否かを判断し、その判断結果に基づき、他の計算機へ共通サービスプロセスを再配置することにより、計算機間の共通サービスへの要求に伴う通信オーバーヘッドを低減させることを特徴とする。

【0014】

【作用】前記手段によれば、クライアントがある計算機へ要求を送った後、返ってくる結果を監視し、その結果内に示されている実際に要求を処理した計算機を認識し、もし要求を送った計算機とは違う別の計算機が処理していたならば、実際に処理をした計算機を記憶しておき、以後クライアント側から同一の要求が発生した場合、記憶してある計算機へその要求を送ることにより、計算機間での要求転送の再発を防ぎ、それに伴う通信オーバーヘッドの発生を防止する。

【0015】

【実施例】以下、図面を参照して本発明の実施例を詳細に説明する。

【0016】図1は、本発明の実施例におけるクライアント・サーバ型分散処理システムの構成を示すブロック図である。

【0017】図1において、クライアントシステム10、計算機20、30及び共通資源50とは、ネットワーク40で接続されており、クライアントシステム10にはクライアントシステム制御プログラム12とクライアントプログラム11とが、また計算機20、30には計算機制御プログラム22、32と、サービス処理プロセス21、31とが存在する。更に、計算機20には共通サービス処理プロセス23が存在する。

【0018】計算機制御プログラム22、32はサービスプロセス及び共通サービスプロセスへの要求を受付け、現在その要求を処理すべきプロセスが処理中であ

4

ば、要求を待たせておくためのキューへ処理が終了するまで格納する。

【0019】初めに、負荷分散時の通信オーバーヘッドの低減についてその動作を説明する。

【0020】クライアントシステム制御プログラム12は、クライアントプログラム11からのサービス要求を受け取ると、サービスと計算機との対応表（後述する）から、そのサービスを提供する計算機（この場合は、計算機20）を決定し、クライアントプログラム11のサービス要求をネットワーク40を介して計算機20の計算機制御プログラム22に送信する。

【0021】計算機制御プログラム22は、クライアントシステム制御プログラム12からのサービス要求を受け取ると、計算機20の起動中のプロセスの数、待ちjob数等の負荷の状態を見て、その要求の処理が可能ならばサービス処理プロセス21を選択して、その要求を処理する。サービス処理プロセス21により実行されたサービスの実行結果は、クライアントシステム制御プログラム12を経由してクライアントプログラム11に返送される。しかし、計算機20の負荷の状態を見て、負荷分散が必要となきには、計算機制御プログラム22は、負荷分散を行い、負荷の小さな計算機30を選択し、クライアントプログラム11のサービス要求を計算機制御プログラム32へ転送する。

【0022】計算機制御プログラム32は、計算機制御プログラム22から転送されたサービス要求を受けると、サービス処理プロセス31を選択して、そのサービス要求を処理する。サービス処理プロセス31により実行されたサービスの実行結果は、計算機制御プログラム32を経由してクライアントシステム制御プログラム12に返送される。

【0023】クライアントシステム制御プログラム12は、先ほどサービス要求を送った計算機20とは違う計算機30から結果が返送されたことを知り、それに対応するサービスと計算機の対応表の項目を更新し、そして、クライアントプログラム11へ結果を返送する。

【0024】クライアントシステム制御プログラム12は、再びクライアントプログラム11から同一のサービス要求を受け取ると、クライアントシステム制御プログラム12は、前回のサービス要求のときと同じ計算機20にクライアントプログラム11のサービス要求を送信せずに、記憶されているサービスと計算機の対応表から今度は計算機30の計算機制御プログラム32にクライアントプログラム11のサービス要求を送信する。

【0025】図2～図4は、本発明の実施例におけるクライアントシステム制御プログラム12、及び計算機制御プログラム22、32の負荷分散時の処理手順のフローチャートである。

【0026】図2において、クライアントシステム制御プログラム12は、クライアントプログラム11からサ

5

サービス要求(ステップ201)を受け取ると、要求されたサービスについて、記憶されているサービスと計算機との対応表を検索し、そのサービスを提供する計算機として計算機20を決定する(ステップ202)。

【0027】クライアントシステム制御プログラム12は、送信メッセージ中の転送メッセージ識別フィールドに負荷分散によって発生した転送メッセージではないことを示す値をセットし(ステップ203)、その送信メッセージを該当する計算機の計算機制御プログラム22にネットワーク40を介して送信する(ステップ204)。

【0028】その後、クライアントシステム制御プログラム12は、計算機制御プログラムからのサービス実行結果メッセージを受信し(ステップ205)、受信メッセージ中の計算機1Dと、サービスと計算機の対応表内の該当項目を比較し(ステップ206)、同一でない場合は、次の同一のサービス要求に備えてサービスと計算機の対応表の計算機の項目を実際に要求を処理した計算機に更新し(ステップ207)、サービス実行結果をクライアントプログラム11に返送する。

【0029】クライアントシステム制御プログラム12は、再びクライアントプログラム11から同一のサービス要求を受け取ったとき、クライアントシステム制御プログラム12は、更新されたサービスと計算機の対応表から、計算機制御プログラム32にサービス要求を送信する。

【0030】図3において、計算機制御プログラム22は、クライアントシステム制御プログラム12からのメッセージを受信すると(ステップ301)、受信メッセージ中の転送メッセージ識別フィールドの値を見て、転送メッセージであるかどうかを判断し(ステップ302)、転送メッセージであれば、後述の転送メッセージ処理を行い(ステップ303)、転送メッセージでなければ、下記のクライアントメッセージ処理に移行する。

【0031】計算機制御プログラム22は、計算機20の負荷状態を調べ、負荷分散が必要か否かを判断し(ステップ304)、負荷分散の必要性がなければ、サービス処理プロセス21を選択し(ステップ305)、サービス要求を処理させる(ステップ306、ステップ307)。サービス要求を受け取ったサービス処理プロセス\*40

6

\*21は、その結果を計算機制御プログラム22に返送する(ステップ308)。サービス結果を受けた計算機制御プログラム22は、計算機1Dをサービス実行結果メッセージの特定のフィールドにセット(ステップ309)した後、ネットワーク40を介して、クライアントシステム制御プログラム12にサービス結果を返送する(ステップ310)。

【0032】また、計算機20の負荷状態より負荷分散が必要となった場合には、計算機制御プログラム22は、軽負荷の計算機を探し、転送先計算機として計算機30を選択し(ステップ311)、クライアントシステム制御プログラム12から受信したメッセージ中の転送メッセージ識別フィールドに、負荷分散に伴う転送メッセージであることを示す値をセット(ステップ312)した後、ネットワーク40を介して、計算機制御プログラム32に転送メッセージを送信する(ステップ313)。

【0033】図4において、計算機制御プログラム32は、計算機制御プログラム22からのメッセージを受信(ステップ401)すると、受信メッセージ中の転送メッセージ識別フィールドの値を見て、転送メッセージか否かを判断し(ステップ402)、転送メッセージでなければ、前述のクライアントメッセージ処理を行う(ステップ409)。

【0034】また、転送メッセージであれば、サービス処理プロセス31を選択し(ステップ403)、サービス要求を処理させる。サービス要求を受け取ったサービス処理プロセス31は、サービス実行後(ステップ404、405)、その結果を計算機制御プログラム32に返送する(ステップ406)。

【0035】サービス結果を受けた計算機制御プログラム32は、計算機1Dをサービス実行結果メッセージの特定のフィールドにセット(ステップ407)した後、ネットワーク40を介して、クライアントシステム制御プログラム12にサービス結果を返送する(ステップ408)。

【0036】本実施例におけるサービス内容と計算機と対応関係を表1に示す。

【0037】

【表1】

サービス内容	計算機20	計算機30
サービスA	○	×
サービスB	○	○
サービスC	○	×

【0038】図5は、クライアントシステム制御プログラム12と計算機制御プログラム22、32、または、計算機制御プログラム22と計算機制御プログラム32の間で、やり取りされる送受信メッセージの構成を示す図である。

【0039】図5に示すように、送受信メッセージ50は、サービス要求を行ったクライアントのクライアントシステムIDフィールド51、クライアントがサービス実行を依頼した、または、実際にサービスを実行した計算機1Dをセットするための計算機1Dフィールド5

7

2、サービスを特定するためのサービス名称フィールド53、計算機間の転送メッセージかどうかの識別をするための値をセットする転送メッセージ識別フィールド54、及びデータを格納するデータフィールド55から構成される。

【0040】以上説明したように、本実施例によれば、動的負荷分散環境で、クライアントがサービス要求を送った計算機の負荷の状態により負荷分散が発生し、他の計算機へその要求が転送された場合、クライアントが次回から同一のサービス要求を前回にサービスを実際に実行した計算機へ送ることになるため、計算機間のサービス要求転送に伴う通信オーバーヘッドを防止でき、システム全体の処理性能の低下を防止できる。

【0041】次に、共通サービスへの要求時の通信オーバーヘッドの低減についてその動作を説明する。

【0042】計算機20内に存在する共通サービス処理プロセス23（共通サービス処理プロセス1）は、計算機制御プログラム22からの要求を受け取ると、共通サービスの処理を実行し、計算機20、30からの共通サービスへの要求頻度を求め、再配置するか否かを判断し、必要ならば再配置を行う。

【0043】図6は、共通サービス処理プロセス1が移動元の計算機20で行う処理手順のフローチャートである。

【0044】図6において、共通サービス処理プロセス1は、計算機制御プログラム22からの共通サービス処理要求を受信すると（ステップ601）、共通サービスの処理を行った後（ステップ602）、各計算機20、30からの共通サービスの利用頻度を求め（ステップ603）、共通サービス処理プロセス1の利用頻度が、共通サービス処理プロセス1の存在する計算機よりも他の計算機の方が一定以上大きい場合には、共通サービス処理プロセス1の存在する計算機よりも他の計算機の方が一定以上大きい場合には、共通サービス処理プロセス1の利用頻度が、共通サービス処理プロセス1の存在する計算機よりも他の計算機の方が一定以上大きい場合には、共通サービス処理プロセス1は、計算機制御プログラム22、ネットワーク40を介して、他の計算機（この場合は計算機30）に対して、共通サービス処理プロセス2の起動を要求する（ステップ605）。

【0045】共通サービス処理プロセス1の利用頻度が、共通サービス処理プロセス1の存在する計算機よりも他の計算機の方が一定以上大きい場合には、共通サービス処理プロセス1の存在する計算機よりも他の計算機の方が一定以上大きい場合には、共通サービス処理プロセス1は、計算機制御プログラム22、ネットワーク40を介して、他の計算機（この場合は計算機30）に対して、共通サービス処理プロセス2の起動を要求する（ステップ605）。

【0046】その後、計算機制御プログラム22からの処理要求を受信すると（ステップ606）、受信した処理要求が共通サービス要求か否かを判断し（ステップ607）、共通サービス処理の要求であれば、計算機制御プログラム22からの終了要求メッセージを受信するまで共通サービスの要求を処理する。

【0047】受信した処理要求が、終了要求メッセージ

8

であれば、計算機30内に起動した共通サービス処理プロセスに対して、終了メッセージを送信し（ステップ609）、共通サービス処理プロセスを終了する（ステップ610）。

【0048】図7は、計算機制御プログラム22の処理手順のフローチャートである。

【0049】図7において、移動元の計算機制御プログラム22は、メッセージを受信すると（ステップ701）、受信したメッセージが共通サービス要求か否かを判断する（ステップ702）。

【0050】受信したメッセージが共通サービス要求であれば、要求内容に要求元の計算機のIDを付加し、共通サービス処理プロセス1に処理を要求する。

【0051】また、受信したメッセージがサービス要求でない場合には、受信したメッセージが起動完了のメッセージか否かを判断する（ステップ705）。

【0052】起動完了のメッセージを受信したのであれば、共通サービスの要求の受け付けを停止し（ステップ706）、他の計算機制御プログラム32へ共通サービス要求の受け付けの開始要求を送信し（ステップ708）、さらに、共通サービス処理プロセス13へ終了要求を送信する（ステップ708）。

【0053】図8は、計算機制御プログラム32の処理手順のフローチャートである。

【0054】図8において、計算機制御プログラム32は、メッセージを受信すると（ステップ801）、受信したメッセージが共通サービス処理プロセスの起動要求か否かを判断する（ステップ802）。

【0055】共通サービス処理プロセスの起動要求を受信したのであれば、計算機内に共通サービス処理プロセス2の起動を開始し（ステップ803）、ネットワーク40を介して、計算機制御プログラム22からの共通サービス要求受け付けの開始要求を受信したら（ステップ804）、共通サービス要求の受け付けを開始する（ステップ805）。

【0056】図9は、計算機30内に新たに起動する共通サービス処理プロセス33（共通サービス処理プロセス2）の処理手順のフローチャートである。

【0057】図9において、計算機制御プログラム32のサービス処理プロセス2の起動開始に対応して、共通サービス処理プロセス2の起動を開始し（ステップ901）、起動が完了すると（ステップ902）、計算機制御プログラム22に対して、起動完了のメッセージを送信する（ステップ903）。

【0058】その後、計算機20内の共通サービス処理プロセス1からのプロセス終了メッセージを受信すると（ステップ904）、共通サービスの処理を開始し（ステップ905）、以後共通サービス処理プロセス1と同一の処理を行う（ステップ906）。

【0059】共通サービス処理プロセスの起動は、潜在

的に持たせたものを活性化するか、あるいは、計算機の磁気ディスクからロードする等の方法で起動される。

【0060】本実施例によれば、複数の計算機をネットワークで接続し、各計算機にかかる負荷を監視し、負荷のバランスを保つ動的負荷分散環境下で、各計算機を呼び出して1つの処理を実行していくクライアント・サーバ型分散処理システムにおいて、共通サービス処理プロセス自身が効率的な良い計算機を判断し、共通サービス処理プロセスの再配置を行うので、計算機間の共通サービスの要求に伴う通信オーバーヘッドを低減でき、システム全体の処理性能の向上を図ることができる。

【0061】以上、本発明を実施例に基づき具体的に説明したが、本発明は、前記実施例に限定されるものではなく、その要旨を逸脱しない範囲で種々変更し得ることは言うまでもない。

#### 【0062】

【発明の効果】以上説明したように、本発明によれば、複数の計算機をネットワークで接続し、各計算機にかかる負荷を監視し、負荷のバランスを保つ動的負荷分散環境下で、クライアントがサービス要求を送った計算機の負荷の状態により負荷分散が発生し、他の計算機へその要求が転送された場合、クライアントが次回から同一のサービス要求を前回にサービスを実際に実行した計算機へ送ることになるため、計算機間のサービス要求転送に伴う通信オーバーヘッドを防止でき、システム全体の処理性能の低下を防止できる。

【0063】また、本発明によれば、複数の計算機をネットワークで接続し、各計算機にかかる負荷を監視し、負荷のバランスを保つ動的負荷分散環境下で、各計算機を呼び出して1つの処理を実行していくクライアント・サーバ型分散処理システムにおいて、共通サービス処理プロセス自身が効率的な良い計算機を判断し、共通サービス処理プロセスの再配置を行うので、計算機間の共通サービスの要求に伴う通信オーバーヘッドを低減でき、システム全体の処理性能の向上を図ることができる。

#### 【図面の簡単な説明】

【図1】本発明の実施例におけるクライアント・サーバ型分散処理システムの構成を示すブロック図である。

【図2】本発明の実施例におけるクライアントシステム制御プログラムの負荷分散時の処理手順のプロチャートである。

【図3】本発明の実施例における計算機制御プログラム22の負荷分散時の処理手順のプロチャートである。

【図4】本発明の実施例における計算機制御プログラム32の負荷分散時の処理手順のプロチャートである。

【図5】本発明の実施例におけるクライアント・サーバ型分散処理システムにおいて使用される送受信メッセージの構成を示す図である。

【図6】本発明の実施例における共通サービス処理プロセス1の処理手順のプロチャートである。

【図7】本発明の実施例における計算機制御プログラム22の共通サービス時の処理手順のプロチャートである。

【図8】本発明の実施例における計算機制御プログラム32の共通サービス時の処理手順のプロチャートである。

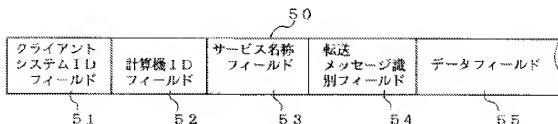
【図9】本発明の実施例における、新たに起動する共通サービス処理プロセス2の処理手順のプロチャートである。

#### 【符号の説明】

10…クライアントシステム、11…クライアントプログラム、12…クライアントシステム制御プログラム、20…計算機、21…サービス処理プロセス、22…計算機制御プログラム、23…共通サービス処理プロセス、30…計算機、31…サービス処理プロセス、32…計算機制御プログラム、40…ネットワーク、50…送受信メッセージ、51…計算機IDフィールド、52…計算機IDフィールド、53…サービス名称フィールド、54…転送メッセージ識別フィールド、55…データフィールド。

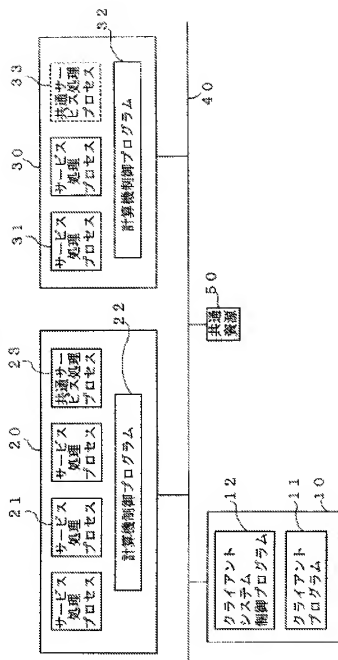
【図5】

図5



【図1】

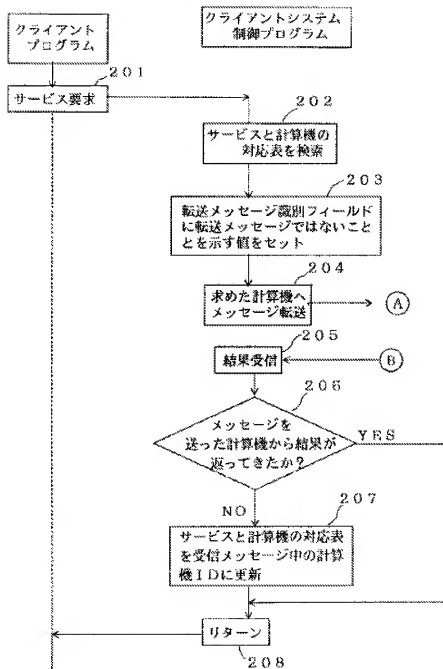
図1





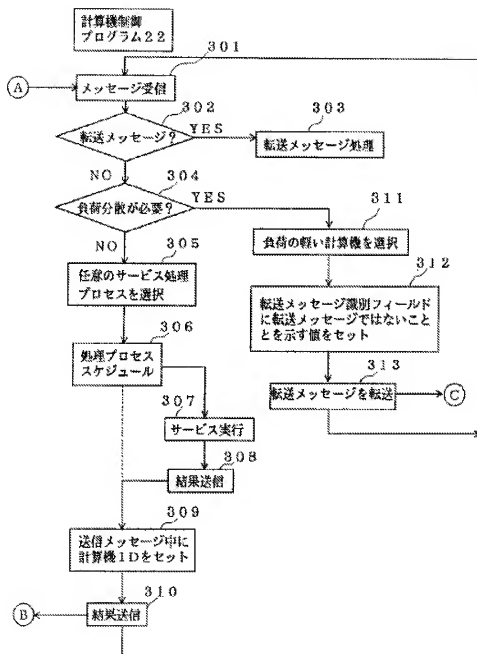
【図2】

図 2



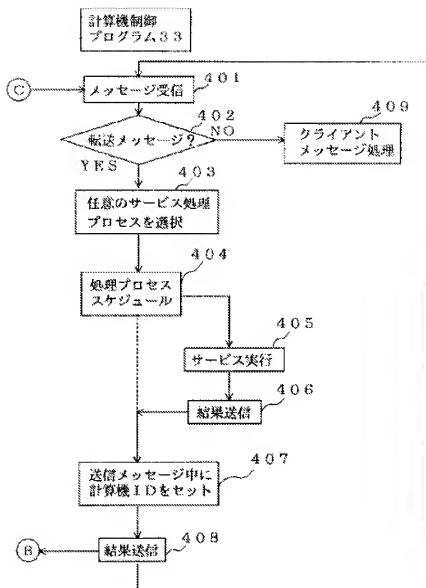
【図3】

図 3

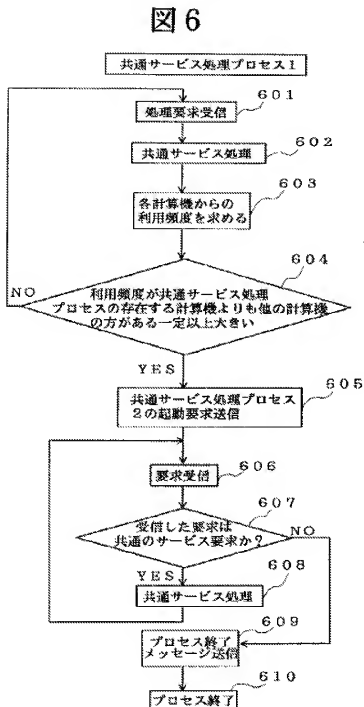


【図4】

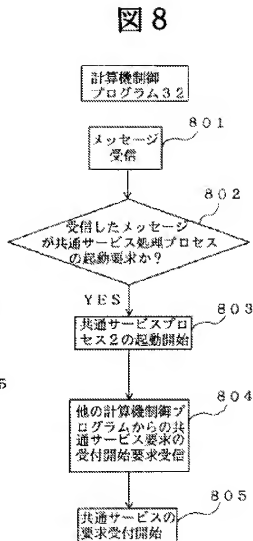
図 4



【図6】

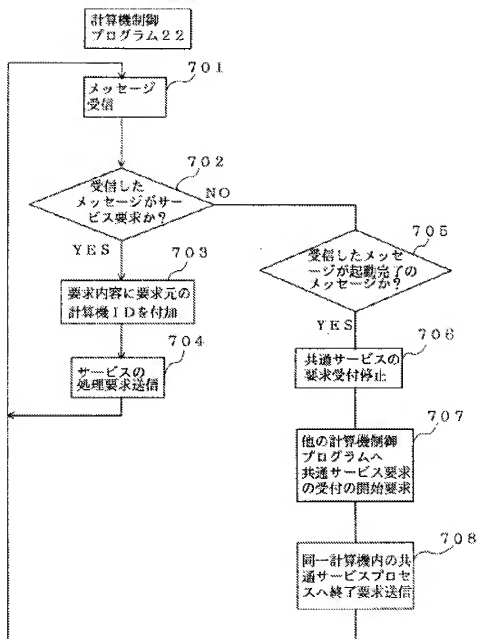


【図8】



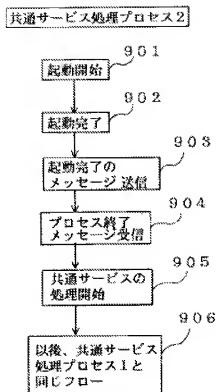
【図7】

図7



【図9】

## 図 9



フロントページの続き

(72)発明者 長谷川 宏行  
神奈川県横浜市戸塚区戸塚町5030番地 株  
式会社日立製作所ソフトウェア開発本部内

(72)発明者 橋本 忠士  
神奈川県横浜市戸塚区戸塚町5030番地 株  
式会社日立製作所ソフトウェア開発本部内